V1.2 - 6 February 2021

A WiFi Antenna Rotator Compass

Giovanni Carboni, IZ5PQT
Via Battelli 11, 56127 PISA (Italy)
iz5pqt@gmail.com

Did the encoder on your rotary beam break down and you don't know where your antenna is pointing? Do you want to add a precise direction readout to your field day directive beam or Yagi? Then build this magnetic compass that will show on a smartphone the precise aiming of your antenna. It only requires two wires for power and, if operating /P, an inexpensive 5V power bank for cellphones will operate your compass totally wireless for a day.

The compass uses two basic ingredients: the Honeywell HMC5883L magnetic field sensor and the Arduino MKR WiFi 1010 microcontroller board. Just four wires suffice to interconnect the two.

HMC5883L

To quote the manufacturer datasheet "the Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as lowcost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I2C serial bus allows for easy interface." I recommend to purchase a breakout board to avoid manipulating the 3x3x0.9 mm SMD package. I used one made by Olimex https://www.olimex.com/Products/Modules/Sensors/MOD-HMC5883L/open-source-hardware but the Adafruit board is perfectly equivalent and in fact I used the Adafruit libraries for Arduino in the project.

Arduino MKR WiFi 1010

The magnetic sensor sends its data to Arduino via the I2C serial bus. Arduino decodes the Earth's magnetic field data to provide the compass readout. The Arduino board used here is the MKR 1010, based on a powerful 32-bit ARM microcontroller and having a built-in WiFi module with antenna. The MKR 1010 can be easily setup as a Telnet server, using existing software. Interfacing to the HMC5883L is extremely simple, via two I2C serial lines. The MKR 1010 board is powered with 5 V but the input/output signals must be strictly 3.3 V max, the same voltage levels of the magnetic sensor. The Vcc and GND pins of the Arduino board provide the 3.3 V supply to the magnetic sensor. The circuit draws 115 mA. The schematic diagram is shown in Fig. 1.

Hardware

The compass is mounted inside a waterproof electrical junction box, approximately

10x10 cm and 5 cm deep (see Fig. 2). The box must be made of plastic to avoid shielding the WiFi signal. The two boards are attached to the inside of the cover plate. The Arduino board plugs into two female header strips whose pins are soldered to a scrap breadboard, fastened with hot glue to the cover. A piece of double-sided tape keeps the HMC5883L board in place. In my case the HMC5883L is mounted upside down if the cover plate is above. This requires reversing one of the axes in the software decoding routine. The x,y,z axes are silkscreen printed on the sensor board. Heading is referred to the y-axis of the sensor. Try to align the axis with a side of the box. Misalignments can be easily corrected in the software, anyway. See Fig. 3 for details.

The interconnection of the Arduino and compass boards requires only 4 wires: Vcc, GND, SDA and SCL (all clearly marked on the Arduino pins). I did not do any soldering: I used small jumper wires. The female header strips where the Arduino board plugs in are dummy. All the connections are made directly to the board itself. Before attaching the components to the cover make sure that the USB connector on the MKR1010 board is not obstructed.

The DC supply is brought in via a SO-239 socket mounted on the box wall. The reason for the bulky connector is simply that is reasonably rugged for outdoor use.
A 7805 regulator inside the box can be switched out and in to allow operation either at 5 V, using a cell phone power bank, or with an external supply, even unregulated. The 7805 needs a small heathsink since at 12 V it dissipates about 0.8 W.

The box is screwed onto a L-shaped aluminum plate (10x15 cm). An U-bolt is attached to the 5 cm vertical lip to allow fastening the box to the antenna mast, but you can use the method you prefer (Fig. 4). Just remember not to use iron or steel for the plate in order not to upset the Earth magnetic field and consequently the compass reading. See photo.

Software

The program (or sketch in Arduinese) `MKR_Telnet_Compass.ino` is just a cut-and-paste of two existing pieces of software. The credits are in the comments. The program `setup` section configures and starts the Telnet server. Once the server is fired up the code obtains the compass data from the routine `getheading()`and writes them to the server for transmission in an infinite loop. To install the sketch open the Arduino IDE on your PC. Go to the Board Manager and search for the package "ArduinoAVR Board". Install it. Then install the package "Arduino SAMD (32-bits ARM Cortex-M0+) Boards". At this point the Menu will propose several boards to you: choose "Arduino MKR WiFi 1010". Connect the Arduino to your PC via an USB cable and select the proper COM port that should now be visible. Then open the sketch `MKR_Telnet_Compass.ino` and via the library manager install the following libraries: "WiFiNINA", "Adafruit HMC5883 Unified" and "Adafruit Unified Sensor". You must now configure the WiFi network in the sketch. In the IDE you will notice the `arduino_secrets.h` tab. Click on it and choose a name for your WiFi network. I use mkr-net and no password. Then load the sketch in the Arduino. If you wish you can see some debug information in the IDE serial monitor. Then scan for local WiFi networks with your smartphone and check that mkr-net is actually on air. The IP address should be by default 192.168.4.1 port 23. The USB cable can now be disconnected.
The sketch allows to correct the compass reading for the magnetic declination. This can also be used to correct any misalignment of the antenna w.r.t. the box or the HMC5883 board.

Usage

The compass heading is transmitted over the network using the Telnet protocol. In order to display it your phone WiFi must be connected to mkr-net and have a Telnet terminal app installed. For Android there are several choices on the Play Store, all more or less equivalent. I use Termius. For the iPhone there is iTerminal. Configure the terminal app to use Telnet (not ssh) connecting with the IP address 192.168.4.1 port 23. Once you hit “connect” you will see the compass heading scrolling continuously on the screen (Fig. 5). Notice that your phone must be connected to mkr-net during use, so you may wish to have an old cellphone dedicated to the compass.

I tested the compass on the mast of a 3-element HF beam and its operation was completely satisfactory. The maximum power used was 200 W and no adverse RFI effect was noted. But be warned: with higher powers your mileage may vary.

The MKR1010 WiFi has a range of about 300 ft outdoors. This is plenty of coverage for /P operation. Notice that the power cannot be adjusted and no external antenna can be connected. However if the receiving smartphone is inside a building (in particular a brick or concrete building) the range will be reduced. In this case I had very good success with a WiFi range extender, strategically placed in sight of the compass and of the shack.

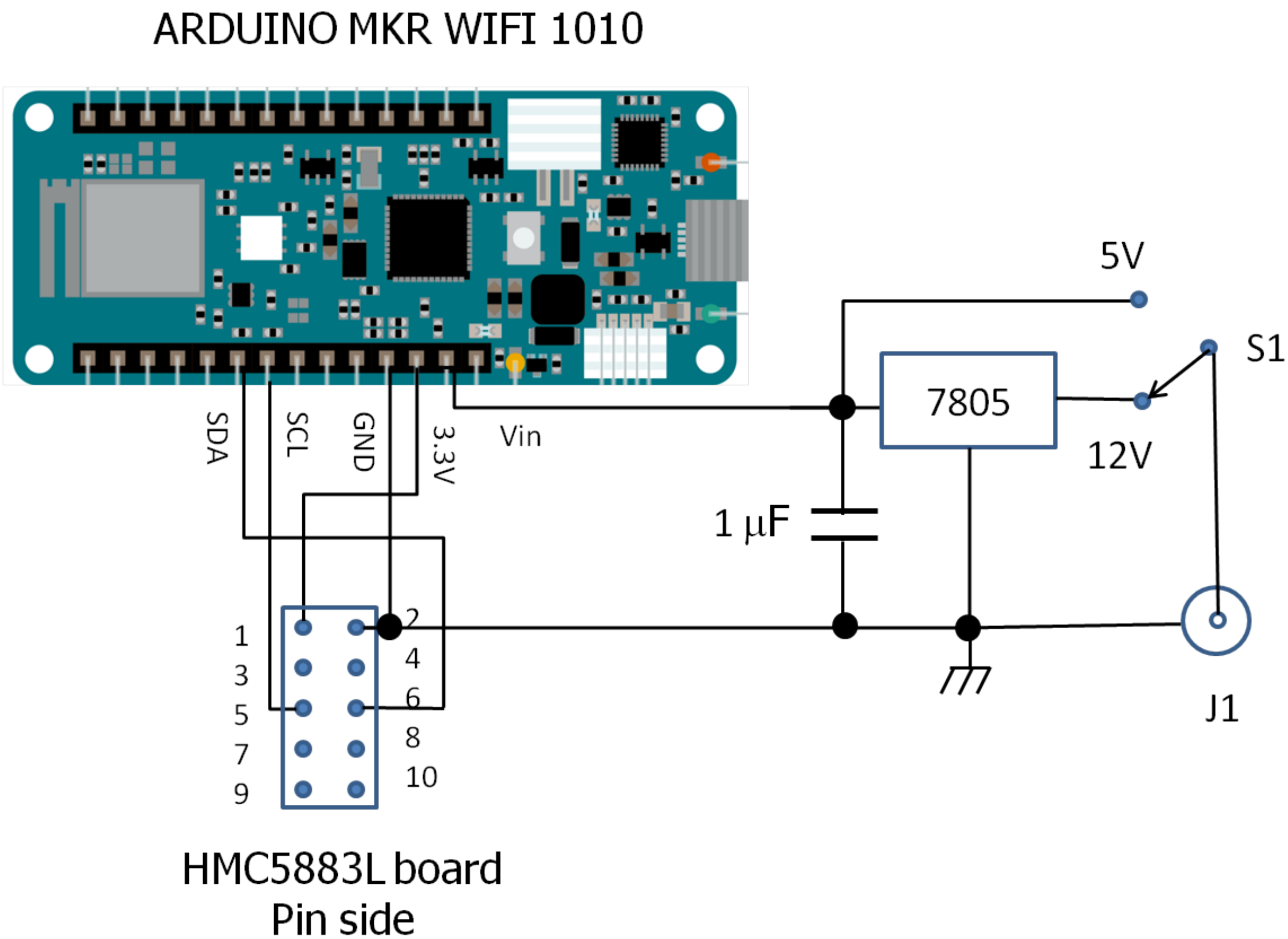

Fig. 1 Schematic diagram. The HMC5883L connections refer to the Olimex breakout board (see text).

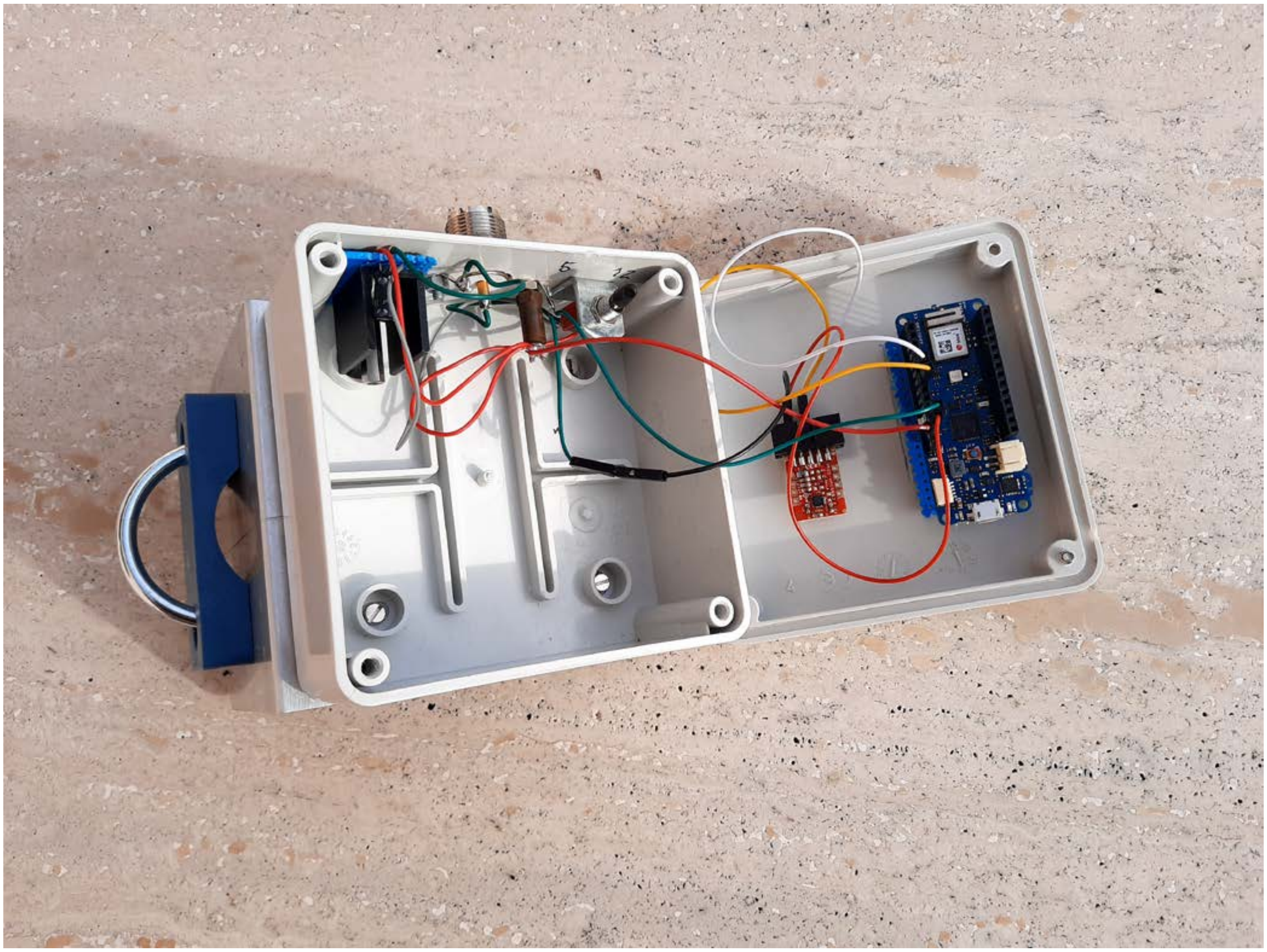

Fig. 2 The circuit inside the junction box. The Arduino and the magnetic sensor boards are attached to the cover. Inside the box are the 7805 regulator with heathsink and the toggle switch to select the supply voltage. The SO-239 is the DC supply connector.

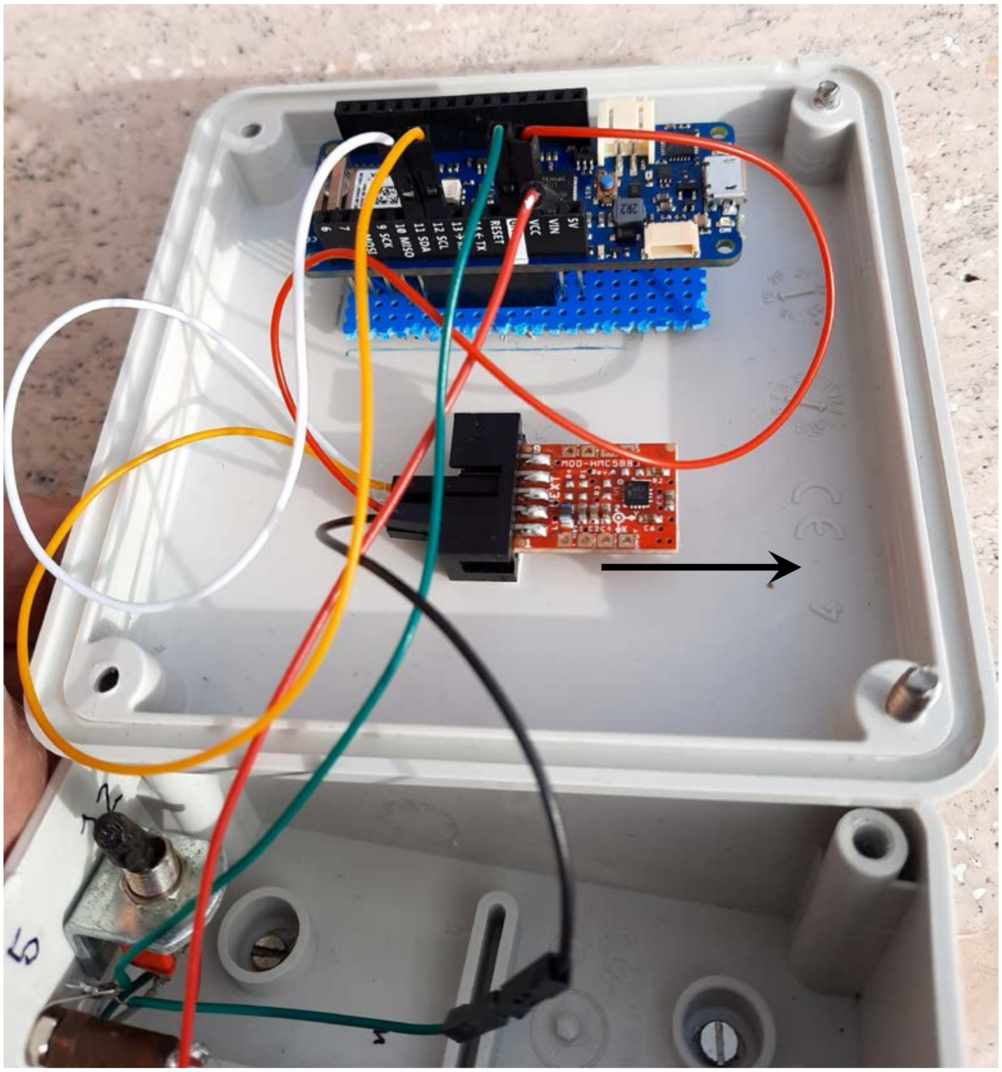

Fig. 3 The Arduino board, mounted on a piece of scrap perforated board, and the MC5883L breakout board (bottom), are supported inside the box cover. The arrow points in the direction of the compass "needle". The toggle switch to select the power supply is visible inside the box on the left.

Fig. 4 Mechanical mounting

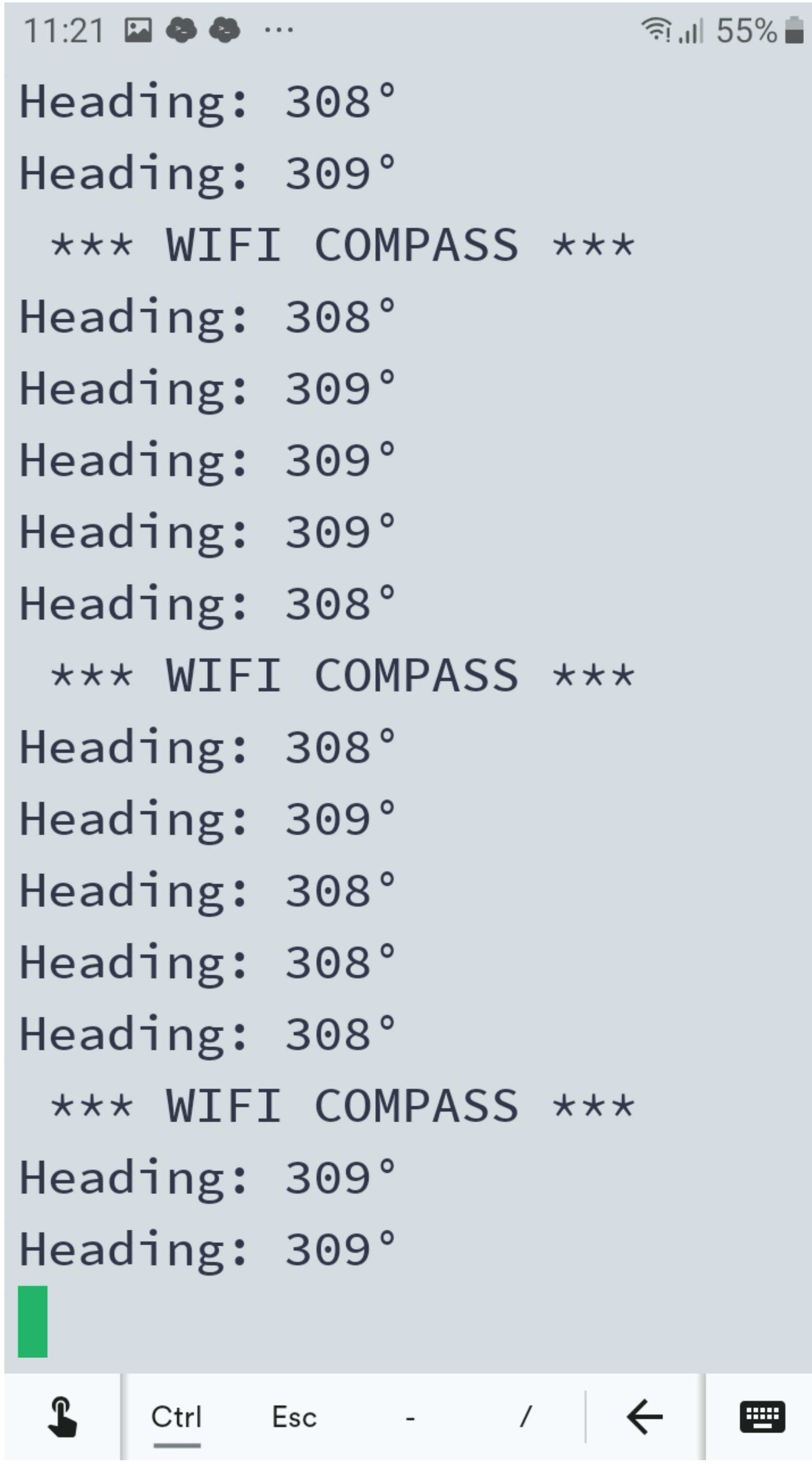

Fig. 5 Compass headings displayed on the smartphone screen (using the Termius terminal app)